

REMARKS

Claims 1, 11, 12, 21, 22, 27, and 31 have been amended to clarify the subject matter regarded as the invention. Claims 1-31 are pending.

The Examiner has rejected claims 1-31 under 35 USC § 103(a) as being unpatentable over CORBA and ORB (Orfali et al., OMG, Iona, Mowbray et al, and Rymer) in view of Yang et al.

The rejection is respectfully traversed. Claim 1 recites “dynamically creating a mapping of said client to said server at run time in response to a request from said client for a connection to said server...wherein dynamically creating the mapping includes determining interface requirements at run time without requiring creation prior to run time of a static interface that defines communication between said client and said server”. An ORB, such as a CORBA ORB, provides a static means of communication across object models and is not capable of dynamically passing a message from one object model to a separate object model without prior knowledge of the desired object model or method. ORBs typically use an IDL to specify the interfaces of client and server objects. Specifically, an IDL is used to pre-define a mapping prior to run time between the client and server objects, e.g., by the creation of static client and server IDL stubs. Such pre-defined mappings may be stored in an interface repository and may be used in conjunction with a dynamic invocation interface. Thus, with a typical ORB, such as an ORB conforming to CORBA, a mapping between a client and a server must be defined before run time and is not dynamically created at run time as recited in claim 1. Claim 1 describes “dynamically communicating an object message between a client and server of separate object models.” Yang et al. do not disclose communication between separate object models. Rather, Yang et al. describe communication between an object-oriented application (CORBA) and a procedural application (DCE). Yang et al. describe invoking a bridge factory to create dynamically or on-demand a bridge between CORBA and DCE applications. Such a bridge factory, however, simply automates the code-generation of a static bridge, and the associated mapping rules must be coded into the bridge factory prior to run time. Yang et al. specifically describe a prototype of a static CORBA-DCE bridge. Thus, Yang et al. do not describe dynamically creating a mapping of a client and server of separate object models at run-time as described in claim 1. Since none of the cited references either separately or combined describe dynamically creating a mapping of a client and server of separate object models at run-time without requiring creation prior to run

time of a static interface that defines communication between the client and server, it is believed that claim 1 is allowable.

Claims 2-11 depend from claim 1 and are believed to be allowable for the same reasons described above.

Claim 12 recites “dynamically handling an object message between a client and server in separate object models” and “dynamically create a mapping of said client to said server at run time in response to a request from said client for a connection to said server...wherein to dynamically create the mapping includes to determine interface requirements at run time without requiring creation prior to run time of a static interface that defines communication between said client and said server”. Therefore, it is believed that claim 12 is also allowable for the same reasons described above.

Claims 13-21 depend from claim 12 and are believed to be allowable for the same reasons described above.

Claim 22 recites “a first object running in said first object model...a second object running in said second object model” and “dynamically create a mapping between said first object and said second object at run time in response to a request from said first object for a connection to said second object and wherein to dynamically create the mapping includes to determine interface requirements at run time without requiring creation prior to run time of a static interface that defines communication between said first object and said second object”. Therefore, it is believed that claim 22 is also allowable for the same reasons described above.

Claims 23-26 depend from claim 22 and are believed to be allowable for the same reasons described above.

Claim 27 recites “dynamically communicating an object message between a client and server in separate object models” and “dynamically creating a proxy object and a stub object at run time in response to a request from said client for a connection to said server...wherein dynamically creating said proxy object and said stub object comprises determining interface requirements at run time without requiring creation prior to run time of a static interface that defines communication between said client and said server”. Therefore, it is believed that claim 27 is also allowable for the same reasons described above.

Claims 28-30 depend from claim 27 and are believed to be allowable for the same reasons described above.

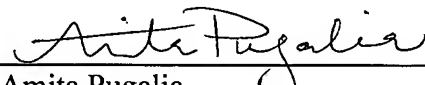
Claim 31 recites “dynamically handling a message between separate object models” and “dynamically creating a mapping of said client to said server at run time...wherein dynamically creating the mapping includes determining interface requirements at run time without requiring creation prior to run time of a static interface that defines communication between said client and said server”. Therefore, it is believed that claim 31 is also allowable for the same reasons described above.

The foregoing amendments are not to be taken as an admission of unpatentability of any of the claims prior to the amendments.

Reconsideration of the application and allowance of all claims are respectfully requested based on the preceding remarks. If at any time the Examiner believes that an interview would be helpful, please contact the undersigned.

Respectfully submitted,

Dated: 10-4-07


Amita Pugalia
Registration No. 57,367
V 408-973-2580
F 408-973-2595

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014